APPENDIX A

Procedures for Medium Access Control in the PL Network System

TABLE OF CONTENTS

1 Introduction

This document presents a proposed Medium Access Control layer intended to be used with a power line networking physical (phy) layer. It assumes that a number of modifications will be made to the proposed physical layer design. These include:

- Changing the packet format so that a contention mode packet begins with a preamble, followed by a ROBO-like header, followed by a payload section which can be modulated with BPSK, QPSK, or ROBO. It is assumed that there are no multiple PPDU transmissions.

- Adding the capability for reservation based transmissions in which reservation packets do not require a ROBO mode header.

2 Procedures for Medium Access Control

This section describes procedures for Medium Access Control (MAC) layer protocol of the powerline networking system. Formats of management messages used to implement the protocol are described in section 3.

The MAC layer of the powerline (PL) network uses a Carrier Sense Multiple Access (CSMA) protocol with modifications to support special requirements for applications requiring low latency. The protocol supports both contention based access and reservation based access. Reservation based access can operate in either a controller-less mode or in a mode with a network controller.

The MAC layer procedures described in this section make use of the management and data packets defined in section 3.

2.1 Procedures for contention based access

The state machine for the MAC of a transmitting client using contention based access is shown in Figure 1. The processing that occurs in each state is described in the following subsections.

Much of the movement between states in this machine is based on the criterion of the availability of the physical layer medium. For the purposes of this state diagram description, "medium unavailable" is intended to denote any of three events:

- The physical medium is presently being used by another client (carrier sense is active), or
- The physical medium is reserved for use by another client at some time in the near future, and the packet that is ready for transmission is long enough that sending it will interfere with the reservation, or
- A client running an upgraded protocol has declared a blanking period at some time in the near future, and the packet that is ready for transmission is sufficiently long that sending it will interfere with the blanking period. The blanking period is used to prevent v1.0 clients from accessing the physical medium for a period of time so that clients running a different protocol can access the medium without interference from the v1.0 clients.

Because this protocol provides the ability to segment data frames, the latter two situations should arise only when a packet is queued for transmission so close to the time of a reservation or a blanking period, that even a fragment of a packet can not be sent.

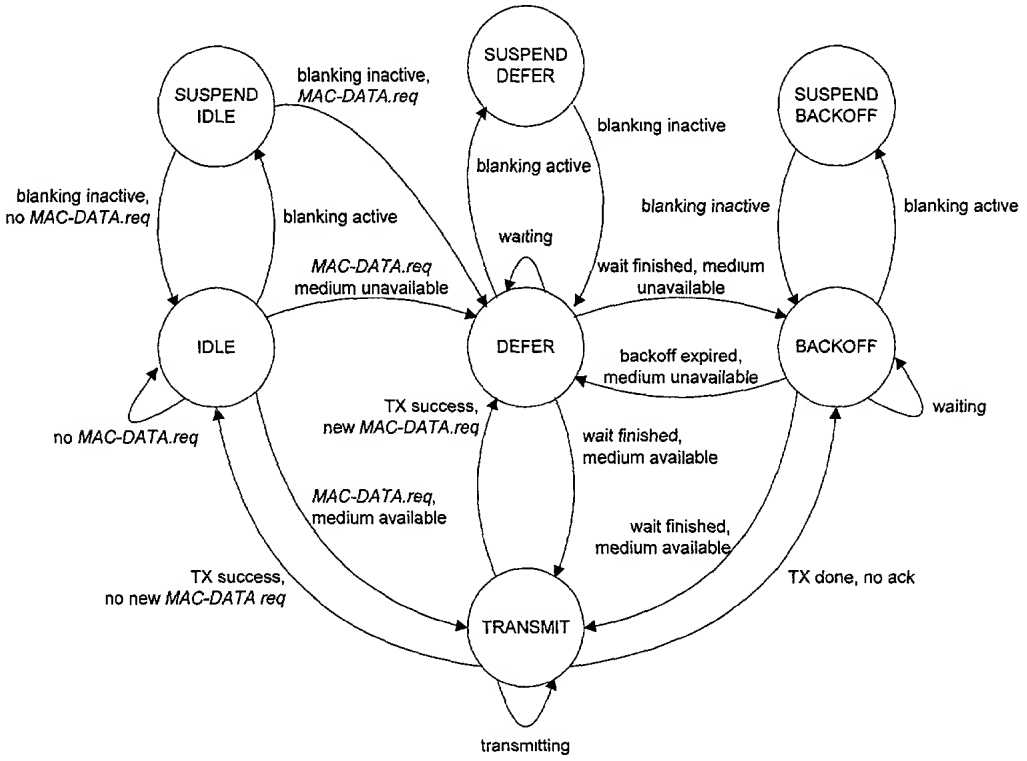

Figure 1 MAC layer state machine with contention based access

2.1.1 IDLE state

In the IDLE state, the MAC is waiting to receive a packet for transmission from the higher layers. The packet is transferred to the MAC via the *MAC-DATA.req* primitive.

If no *MAC-DATA.req* is received and the physical medium is not blanked, the MAC remains in the IDLE state. Upon receipt of a *MAC-DATA.req* primitive, the MAC determines whether or not the physical layer medium is available. If it is, the MAC enters the TRANSMIT state and begins to transmit the packet. If the medium is unavailable, the MAC enters the DEFER state.

If a blanking period becomes active, the MAC transitions to the SUSPEND IDLE state.

2.1.2 TRANSMIT state

In the TRANSMIT state, the MAC transmits the queued packet. At the conclusion of transmission, the intended recipient of the packet acknowledges successful receipt of the packet by transmitting in a time slot reserved for it immediately following the received packet. If the transmitting client receives this acknowledgment it assumes that no collision occurred and that the transmission was successful. If the transmission succeeded and no new *MAC-DATA.req* has been received during the transmission of the packet, it returns to the IDLE state. If the transmission succeeded and a new *MAC-DATA.req* was received, then the MAC transitions to the DEFER state.

In either case, when a transmission succeeds, the MAC state machine sets its backoff window length to zero (see the discussion of the BACKOFF state below).

If no acknowledgment is received, a collision is assumed and the MAC enters the BACKOFF state.

2.1.2.1 Acknowledgment

The time immediately following the transmission of a unicast contention based access packet is reserved for the transmission of an acknowledgment packet by the destination (see section 3.3). The duration of the acknowledgment packet varies depending on the tone mask in use by the network, but is known to all clients. A client transmitting an acknowledgment must begin transmission of the acknowledgment within 128 microseconds after the end of the packet that it is acknowledging.

2.1.3 SUSPEND IDLE state

In the SUSPEND IDLE state, the MAC is waiting for the end of a blanking period. When the blanking period ends, if there is no pending *MAC-DATA.req* then the MAC transitions back to the IDLE state. If there is a *MAC-DATA.req* pending, then the MAC transitions to the DEFER state.

2.1.4 DEFER state

In the DEFER state, the MAC has a packet to transmit, but has sensed the presence of another carrier on the physical medium. It thus must wait for the physical medium to become available. Because other clients may also queue packets for transmission during the period that the physical medium is busy, a contention resolution scheme is used to reduce the likelihood of a collision ensuing immediately after the physical medium becomes available.

Figure 2 shows the timing used in contention resolution. The MAC for a client in the DEFER state chooses an integer n at random with a value from 1 to max_slots (a parameter). The value of n indicates the slot in the contention resolution period in which the client should attempt to transmit.

The parameter max_slots is fixed at 8 when the DEFER state is entered from the states IDLE, TRANSMIT or BACKOFF. When the DEFER state is entered from the states SUSPEND IDLE or SUSPEND DEFER, max_slots is set to the value accompanying the blanking period parameters in either the beacon or medium blanking MAC management payloads (see section 3.1.2.1.1.1). Moreover, an additional contention resolution slot is inserted immediately after the end of the blanking period (labeled x in Figure 3). This contention resolution slot is provided strictly for the use of the non-v1.0 device that has established the blanking timing. In the case that the non-v1.0 device uses this slot, then the contention resolution slot timing used at the end of this transmission is the same as that used at the end of the blanking interval. That is to say, the timing is as shown in Figure 3 and the number of contention resolution slots is determined by the max_slots value in the blanking period parameters.

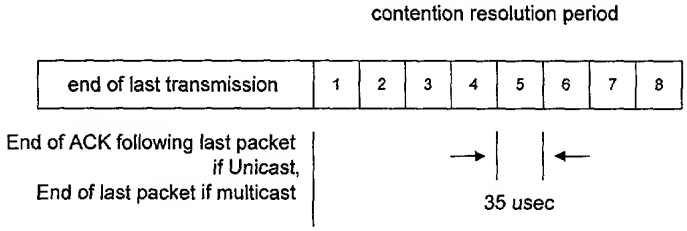

Figure 2 Contention resolution in DEFER state when entered from IDLE, TRANSMIT, or BACKOFF

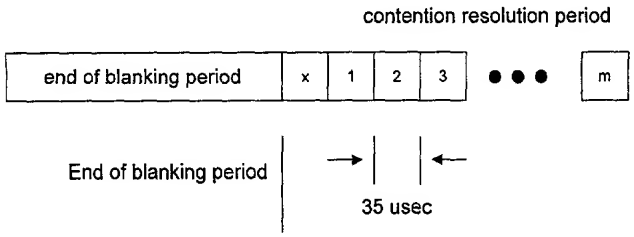

Figure 3 Contention resolution in DEFER state when entered from SUSPEND IDLE or SUSPEND DEFER

When the chosen contention resolution time slot is reached, the MAC first determines whether the physical medium is available. If so, it enters the TRANSMIT state and begins to transmit. If the physical medium is unavailable, the MAC does not initiate transmission but instead enters the BACKOFF state.

At times the MAC layer may receive a *MAC-DATA..req* during the contention resolution period. Assuming that the physical medium is available, then the MAC shall immediately initiate transmission.

If a blanking period becomes active before the chosen contention resolution slot, then the MAC transitions to the SUSPEND DEFER state.

2.1.4.1 Establishing the time of the end of a transmission

In most cases, the time at which a transmission will end can be determined from the length field carried in the ROBO mode header. In some cases, a client receiver may detect a preamble but fail to decode the ROBO mode header. This may be due to a collision in which the colliding packets interfere with each other, or it may be due to an excessively degraded channel. In either case, the MAC layer must assume that the packet has the maximum possible length as specified in section 3.1.1.

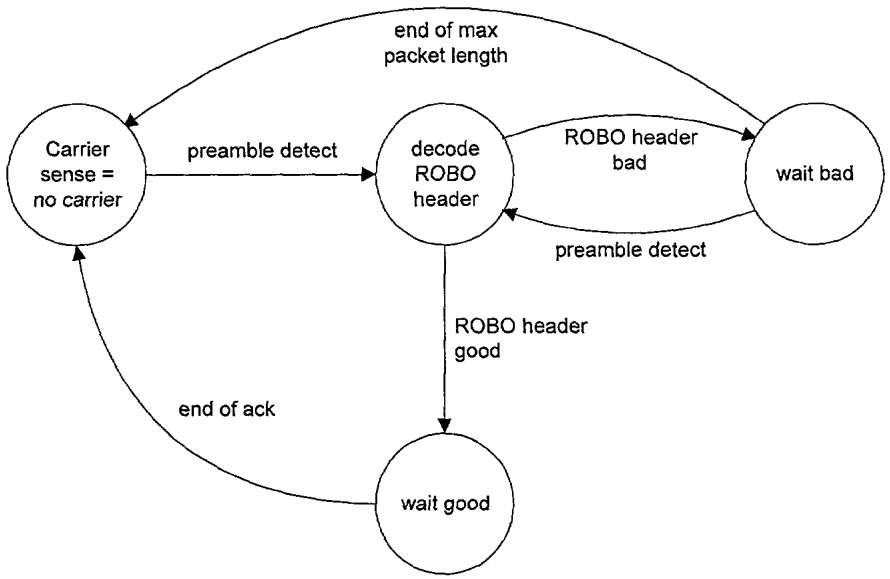

Figure 4 State machine for determining end of packet time

Because many packets are shorter than the maximum length it is possible that another transmission will begin before the end of the waiting period described above. The waiting client thus must continue to try to detect preambles while waiting, and if it does detect a preamble, it should then attempt to receive the ROBO mode header. If the new ROBO header is correctly received, the client then waits according to the length in the new ROBO header. Otherwise it returns to waiting for a period equal to the duration of the maximum length packet plus acknowledgment starting from the beginning of the new preamble. The situation is depicted in Figure 4.

2.1.5 BACKOFF state

The BACKOFF state is used to limit congestion on the physical medium by dispersing the transmit times of clients that have interfered with each others transmit attempts. The MAC layer enters the BACKOFF state in two different circumstances; in the first, a collision is assumed to have occurred because a transmitted packet was not acknowledged. This is called a "collision induced" backoff. In the second case, the client was in the DEFER state, but was pre-empted by another client that chose an earlier contention resolution period in which to transmit. This is called a "defer induced" backoff. Either of these events indicates potential congestion.

The backoff algorithm operates by choosing an integer b at random between 1 and a value called the backoff window length. The MAC state machine then clears and starts a backoff timer, which runs until it expires at a time equal to b times 512 microseconds. On expiration of the backoff timer, if the physical medium is available, the MAC enters the TRANSMIT state and begins transmitting. If the physical medium is not available, the MAC enters the DEFER state.

The MAC includes parameters used to determine the backoff window length. These are tabulated in Figure 5. There are three sets of parameters; one set for high priority clients, another for medium and a third set for low priority clients. The client type is designated by the suffix _h, _m or _l at the end of the parameter name.

Upon entering the BACKOFF state for collision induced backoff, the backoff window length is set to the greater of min_col_backoff or the product of the current backoff window length and col_backoff_factor. If the resulting backoff window length exceeds max_col_backoff, the backoff window length is set to max_col_backoff.

Similarly, upon entry to the BACKOFF state for defer induced backoff, the backoff window length is set to the greater of min_defer_backoff or the product of the current backoff window length and defer_backoff_factor. If the resulting backoff window length exceeds max_defer_backoff, the backoff window length is set to max_defer_backoff.

If a blanking period becomes active prior to the expiration of the backoff timer, the MAC transitions to the SUSPEND BACKOFF state.

If the MAC attempts to transmit a packet more than 8 times without success it must discard all queued packets, set the backoff window length to zero, and return to the IDLE state. For this purpose, a transmission attempt is an entry to the TRANSMIT state.

Parameter	Default value	Meaning
min_col_backoff_h	8	The initial length of the backoff window for high priority

		clients responding to collision
max_col_backoff_h	256	The maximum length of the backoff window for high priority clients responding to collision
col_backoff_factor_h	2	The factor by which the backoff window grows with each collision for high priority clients
min_defer_backoff_h	8	The initial length of the backoff window for high priority clients responding to preemption in defer
max_defer_backoff_h	256	The maximum length of the backoff window for high priority clients responding to pre-emption in defer
defer_backoff_factor_h	2	The factor by which the backoff window grows with each pre-emption for high priority clients
min_col_backoff_m	8	The initial length of the backoff window for medium priority clients responding to collision
max_col_backoff_m	256	The maximum length of the backoff window for medium priority clients responding to collision
col_backoff_factor_m	2	The factor by which the backoff window grows with each collision for medium priority clients
min_defer_backoff_m	8	The initial length of the backoff window for medium priority clients responding to pre-emption in defer
max_defer_backoff_m	256	The maximum length of the backoff window for medium priority clients responding to pre-emption in defer
defer_backoff_factor_m	2	The factor by which the backoff window grows with each pre-emption for medium priority clients
min_col_backoff_l	8	The initial length of the backoff window for low priority clients responding to collision
max_col_backoff_l	256	The maximum length of the backoff window for low priority clients responding to collision
col_backoff_factor_l	2	The factor by which the backoff window grows with each collision for low priority clients
min_defer_backoff_l	8	The initial length of the backoff window for low priority clients responding to pre-emption in defer
max_defer_backoff_l	256	The maximum length of the backoff window for low priority clients responding to pre-emption in defer
defer_backoff_factor_l	2	The factor by which the backoff window grows with each pre-emption for low priority clients

Figure 5 Backoff parameters

2.1.6 SUSPEND BACKOFF state

In the SUSPEND BACKOFF state, the MAC halts the backoff timer for the duration of the blanking period. When the blanking period ends, the MAC transitions back to the BACKOFF state and the backoff timer is allowed to resume running.

2.2 MAC layer beacons and medium blanking procedures

The MAC layer provides the ability for devices designed to an upgraded version of this protocol to restrict the times in which v1.0 compliant devices can transmit. This capability is provided to prevent v1.0 transmissions from interfering with transmissions of non-v1.0 devices, which may follow medium access rules that are not known to v1.0 devices.

The blanking structure is a repeating sequence of times in which v1.0 devices are restricted from contention based access (the "blanking period") and times when they are allowed contention based access (the "v1.0 period"). Reservation based access by v1.0 devices (see section 2.4) is allowed

during the blanking period, but the reservation establishment must be initiated during the v1.0 period.

Figure 6 Blanking structure

The blanking structure is specified by a non v1.0 device. This device transmits a ROBO mode broadcast packet containing a medium blanking payload (see section 3.1.2.1.1.7) which provides a network timing reference and the timing of the blanking period and the v1.0 period. If there are multiple non-v1.0 devices present on the network, these devices must determine which of them is to take responsibility for transmitting the medium blanking information.

The non-v1.0 device must re-transmit the medium blanking payload at least once every five seconds. A special contention resolution slot is provided for the use of the non v1.0 device at the conclusion of each blanking period to ensure that the non-v1.0 device has an opportunity to transmit the blanking information without collision.

Because some clients on the network may be unable to successfully receive the packet containing the medium blanking payload, the MAC protocol includes the capability to propagate the blanking information through the use of beacons. Beacon payloads are transmitted in broadcast packets. Each client transmits beacon packets at a nominal five second rate to indicate its presence in the network and to propagate system timing information. Each client's MAC chooses the exact transmit time for the beacon by selecting an integer t at random between the values of 1 and 1000, and attempting to transmit the packet at a time equal to 4.75 seconds plus t times 500 microseconds since its last beacon transmission. If the physical layer medium is blanked for v1.0 devices at the chosen transmit time then the procedures of section 2.1 must be followed.

If a client receives a medium blanking payload from a non-v1.0 device, it sets the logical distance field of its beacon payload to 0. It sets the beacon fields for duration of the blanking time, duration of the v1.0 time, and max_slots equal to the values received in the medium blanking payload. It computes the current system time by adding the number of microseconds from the start of the first OFDM symbol of the medium blanking message to the start of the first OFDM symbol of the beacon to the system time value in the medium blanking payload.

If the client has not received a medium blanking payload in the last five seconds, then it prepares the contents of its beacon payload using information from the beacon received in the last five seconds with the lowest logical distance. If there are multiple received beacons with the lowest logical distance, then the most recently received beacon shall be used. The client sets the logical distance field of its beacon payload to one more than the logical distance contained in the chosen received beacon. It sets the beacon fields for duration of the blanking time, duration of the v1.0 time, and max_slots equal to the values received in the chosen received beacon. It computes the current system time by adding the number of microseconds from the start of the first OFDM symbol of the chosen received beacon to the start of the first OFDM symbol of the beacon to the system time value in the chosen received beacon payload.

If the logical distance of the chosen beacon exceeds 5, the client assumes that the blanking information is not valid and that the physical medium is continuously available for v1.0 use. It sets the logical distance field of its beacon payload to 7, and it sets the four time fields in octets 2 through 17 to all zeroes.

The packet containing the beacon payload can be used by other clients to assess the link from the transmitting client for the purposes of specifying the payload format to be used on that link. It also can be used to maintain a record of network nodes that each client can communicate with.

2.3 Procedures for network entry

Network entry is the process by which a client having no knowledge of the state of a network gains the knowledge required to allow it to exchange packets with other clients on the network according to the rules for medium access. To enter the network, the client must determine the times at which it may transmit; that is, whether the physical medium is subject to blanking or if there are any reservations active. This requires that the client must remain in receive mode without transmitting for at least five seconds upon initially connecting to the network.

At the end of this five second period, the client may begin to transmit its beacon packet according to the procedures of section 2.2.

The client then may use the procedures of section 2.5 to determine payload formats to be used with each client in the network. Until unique payload formats are negotiated with each client, the entering client must use one of the predefined payload formats to communicate with any other client. When the entering client has determined payload formats for exchanging data with each other client, the network entry process is complete.

2.3.1 Logical Network identifier (LNI)

A client may only be a member of a single logical network. The client can only exchange data with other members of the same logical network. The logical network is identified by a 32 bit LNI field in the beacon payload. A network name of any length may be used by the management entity, but this name is compressed to a 32 bit LNI through the use of a 32 bit CRC as shown in Figure 7. The CRC polynomial represented by this figure is:

$$g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

The LNI is formed as follows. The registers in the CRC are initialized to zero. The two switches are set to the UP position, and the bits in the network name are input into the CRC generator one at a time. When all of the bits in the network name have been input to the CRC generator, the two switches are moved to the down position and the CRC generator is clocked 32 times, producing an output bit at each clock. The first output bit is the lsb of the LNI and the last is the msb.

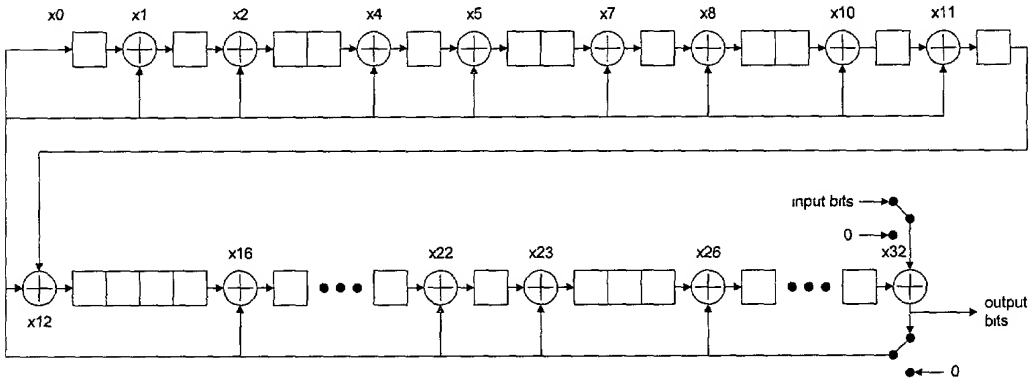

Figure 7 CRC used to generate logical network identifier

Client devices that lack an I/O capability must obtain the LNI from another network device. This occurs in the process of encryption key distribution. The all zero LNI is reserved for use by clients that need to receive the LNI from another client in this way.

2.4 Procedures for controller-less reservation based access

The PL network supports a controller-less reservation based access mode that allows the creation of a virtual circuit connection providing a periodic, low latency, constant bit rate service between an originating client and a destination client.

There are three management related procedures relating to controller-less reservation based access: establishment of the reservation, renewal of the reservation, and termination of the reservation. The maximum reservation duration that can be established has 256 periods or 5 seconds, which ever is smaller. At the end of this time, the reservation must either be renewed or terminated. The renewal process provides the ability to change the payload format in response to changing physical medium conditions.

The management packets used to establish, renew, or terminate a reservation are transmitted as broadcast packets in ROBO mode. Because certain nodes may be able to receive from only one of the two clients involved in the reservation, the reservation information is transmitted by both clients that are party to the reservation.

2.4.1 Reservation establishment

A reservation is established by a handshake process in which the originating client establishes the reservation and the intended recipient acknowledges the establishment.

The originating client begins the process of establishing a reservation by broadcasting a ROBO mode packet containing a reservation establishment (RE) payload. This RE payload informs the other clients in the network of the time at which the reservation is to start, the duration of the packets to be transmitted in the reservation, the period of transmission, and the lifetime of the reservation (that is, the number of packets that will be transmitted during the course of the reservation.) It also provides a capability to establish a reservation for a two way circuit connection by allowing the specification of a duration for a return transmission.

A two way reservation consists of a forward transmission and a reverse transmission, while a one way reservation consists only of a forward transmission. The forward transmission always occurs first, and is transmitted by the originating client. If there is a reverse transmission, it occurs immediately after the forward transmission completes, and is transmitted by the destination client. The reverse transmission must have the same period as the forward transmission, but does not have to have the same duration as the forward transmission. The maximum payload length that may be reserved for either the forward or the reverse transmission is 175 OFDM symbols.

Implicit in the reservation is a time slot in which the destination client can acknowledge the reservation, and a later slot in which the clients can exchange broadcast messages that either terminate or renew the reservation. The time slot reserved for the initial reservation acknowledgment always begins 5 milliseconds after the start of the first OFDM symbol of the preamble of the packet containing the RE payload.

The destination client broadcasts its reservation acknowledgment (RA) payload in a ROBO mode packet, repeating the fields describing the timing of the reservation for the benefit of clients that may have failed to receive the original reservation request.

If the originating client fails to receive the RA payload (which may happen due to a collision with its RE transmission or due to a collision with the RA), it assumes that the reservation has not been established and begins a new establishment procedure.

The timing of the reservation establishment procedure is shown in Figure 8.

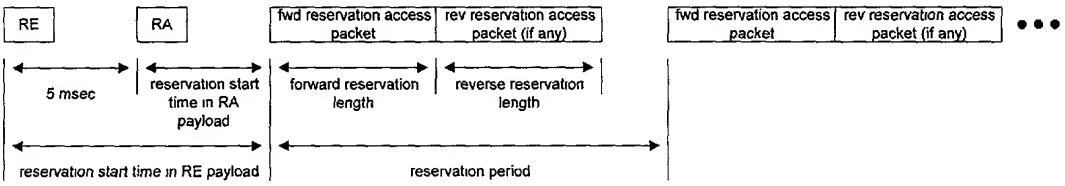

Figure 8 Timing of reservation establishment process

Once the RE and RA have been exchanged, reservation access packets are exchanged according to the agreed upon payload formats. The receiving client does not acknowledge receipt of reservation access packets.

Other clients must not transmit during the reservation time. When a *MAC-data.req* for contention transmission is received by the MAC layer and the physical medium is determined to be available, the MAC layer must ensure that the physical medium will remain available for the duration of the packet to be transmitted. If this is not the case, the MAC must enter the DEFER state and proceed as described in section 2.1.3.

Timing of the reserved slots is differential; that is, each client predicts the start of the next reserved time from the end of the previous reserved time. When multiple reservations are active simultaneously, the client that established the first reservation serves as the reference for all subsequent reservation timing (meaning that the next reserved time for each active reservation is computed relative to the most recent transmission of the first reservation). When the first reservation concludes, the client that established the next reservation becomes the reference. Clients that require different reservation periods from that used by existing reservations must choose a period such that no multiple of that period will result in an overlap of reserved slots.

2.4.2 Reservation renewal and termination

Immediately following the last reservation access period a segment of time is allotted in which a three way handshake is used by the two parties to a reservation to renew or terminate the reservation. This process is shown (for a bi-directional reservation) in Figure 9, and the timing of the messages is shown in Figure 10. The 63 micro-second interval immediately following the last reservation access packet of the reservation is reserved for the client that originated the reservation to begin transmission of a ROBO mode broadcast packet containing a reservation renewal (RR) payload. No other client may attempt to transmit during this 63 microsecond period.

If the reservation is no longer needed, the RR payload terminates the reservation by setting the reservation lifetime field in the RR to 0. Otherwise it renews the reservation by providing new timing parameters for the reservation.

If the reservation is bi-directional and the originating client determines that a different payload format should be used for the reverse transmissions, then it forms the packet with the RR payload to include a PLLC payload with the new parameters to be used by the destination client. The timing information for the reservation must reflect any change in the length of the reservation access packet that will result from the new payload format.

When the destination client receives the RR payload, it responds with another broadcast RR payload. The time for this transmission is reserved so that no other client may transmit during this time and begins 500 microseconds after the conclusion of the last reservation access packet in the reservation.

The destination client sets the reservation lifetime field of its RR payload to agree with that in the RR payload received from the originating client. If the reservation is being terminated (reservation lifetime set to 0) then the handshake is completed once the destination client transmits its RR payload and there is no response from the originating client.

If the reservation is being renewed and the destination client determines that a different payload format should be used for forward transmissions, then it includes a PLLC payload with the new parameters to be used by the originating client in the packet with the RR payload. The destination client updates the reservation timing information to reflect the time needed to support reservation access packets formatted with the new payload format.

If the reservation is being renewed, then when the originating client receives the RR from the destination client it transmits a broadcast RR payload containing the timing information it received from the destination client. The time for this transmission is also reserved (that is, no other client may transmit during this time) and begins 1000 microseconds after the conclusion of the last reservation access packet in the reservation. This completes the handshake, and the reservation access proceeds as before.

If a reservation is active in a network and a client (other than the originating and destination client) fails to receive an RR that terminates or renews the reservation, that client must wait for three more cycles of the reservation before determining that the reservation is inactive.

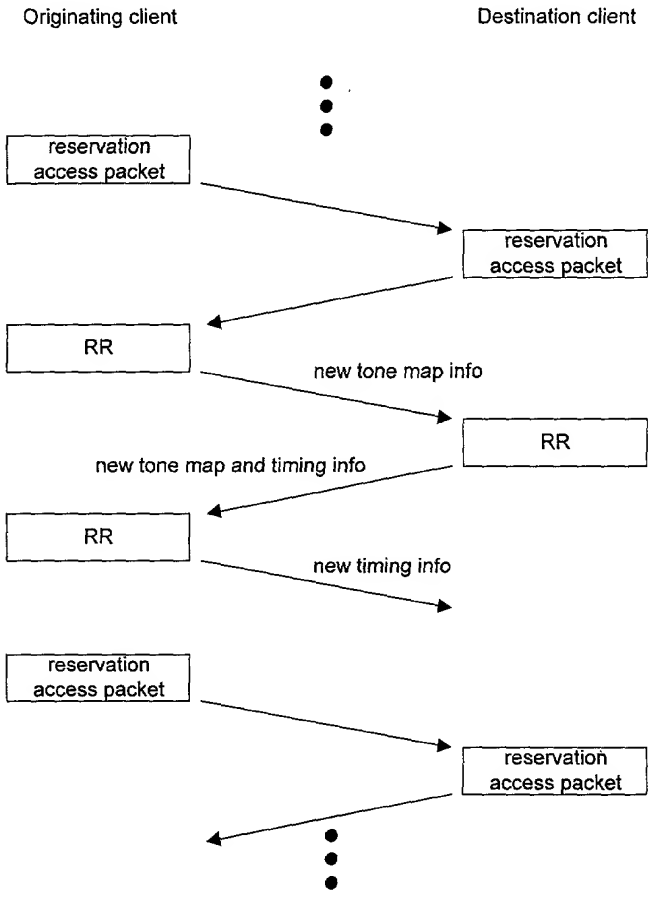

Figure 9 Message flow for reservation renewal

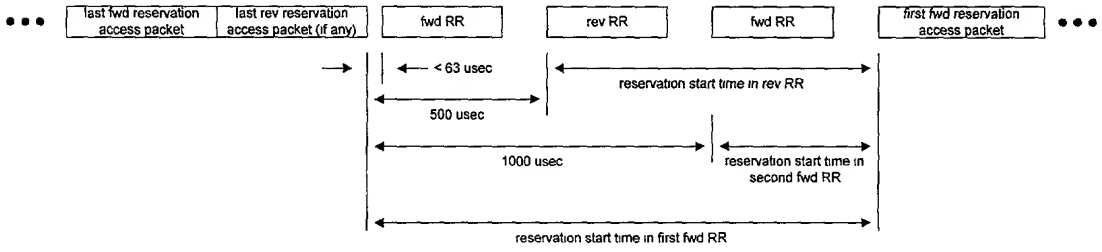

Figure 10 Timing of reservation renewal packets

2.4.3 Error conditions

In some cases, error conditions may make it impossible for the reservation to be established. Either client may reject the information contained in either an RE or an RR payload by setting the status field of the response payload to the appropriate value. Defined values for this field are:

- Status = 0: no error condition

- Status = 1: the reservation is rejected as specified because it interferes with other reservations known to the client sending the status
- Status = 2: the reservation is rejected because the receiver is not ready to accept the data
- Status = 3: a reservation length field exceeds 175 OFDM symbols
- Status = 4: the reservation is rejected for unspecified reasons.

2.5 Procedures for segmentation and reassembly

At times frames received from the network layer may not fit in a single maximum length payload, or they may not fit in the time before the channel becomes unavailable due to blanking or a reservation. The MAC protocol includes provisions for segmentation and reassembly of such frames.

Each unicast data payload includes a sequence number and a segment number. The sequence number is incremented each time a complete frame is successfully transmitted (that is, an acknowledgment is received for each packet carrying a segment of the frame). The segment number is set to zero for unsegmented frames, and for segmented frames it indicates which segment of the frame is being carried in the payload.

If a frame is too large to fit in a single MAC layer packet, then the source client MAC must fragment it into segments that fit into the smallest possible number of MAC layer packets. Alternatively, if a frame is too long to be transmitted in the time interval prior to the beginning of a reservation or a blanking period, the frame must be segmented to allow the greatest portion of the frame possible to fit into the available time. In such a case, sufficient time must be allotted for the acknowledgment to be transmitted prior to the reservation or blanking interval. Frames must be segmented if the first resulting segment can have a payload of 10 or more OFDM symbols. Frames must not be segmented otherwise.

The segment number is initially set to the number of segments in the frame. The packet containing the first segment is transmitted, and if the transmission was successful (as indicated by the receipt of an acknowledgment), the segment number is decremented and the next segment is transmitted.

If an acknowledgment is not received for a segment, the source client must follow the procedures for contention based access of section 2.1 to retransmit the segment. Once the segment is successfully transmitted, the segment number is decremented, and the next segment is transmitted. This process continues until all segments have been successfully transmitted. Upon receiving the payload with segment number set to 1, the destination client can then reassemble the frame.

If the destination client receives a unicast data payload from the same source client with a new sequence number but has not received all the segments of the frame with the previous sequence number, it must discard all the segments of the partial frame.

If the destination client receives an out of sequence payload, an error condition has probably occurred and the destination client is likely to lose encryption synchronization. The destination client can indicate this condition to the source client by transmission of a status payload with error code set to 1.

2.6 Procedures for establishing payload formats

To enable data transfer with the best possible efficiency, each pair of clients must determine the payload format (that is, the payload tone maps, modulation types, and FEC coding) that provides the best throughput on the link connecting them. The means by which the clients determine the best payload format is implementation dependent, but this specification provides features to facilitate the determination.

There are eight pre-defined payload formats. These are used for various types of broadcast messages and to establish communications before any client-defined payload formats are determined.

The test payload and test packet request payload may be used to sound the link between clients using the procedures of section 2.8.

2.6.1 Predefined payload formats

The table of Figure 11 shows the eight predefined payload formats.

Payload format index	C	T	mod	Tone map
0	1/2	8	ROBO	ffff ffff ffff ffff ffff f (all tones used)
1	3/4	8	ROBO	ffff ffff ffff ffff ffff f (all tones used)
2	1/2	4	ROBO	ffff ffff ffff ffff ffff f (all tones used)
3	3/4	4	ROBO	ffff ffff ffff ffff ffff f (all tones used)
4	tbd	tbd	tbd	tbd
5	tbd	tbd	tbd	tbd
6	tbd	tbd	tbd	tbd
7	tbd	tbd	tbd	tbd

Figure 11 Predefined payload formats

2.6.2 Masked tones

Regulatory issues may require that certain tones should not be used in some circumstances. These tones are masked through a *MAC-TONE_MASK.req* primitive and are never used by any client on the network.

2.7 Procedures for encryption

The following are the goals of the PL network security protocol:

- *Confidentiality* – data transmitted on the physical medium is known only to authorized entities. All members of a common network are regarded as authorized entities.
- *Secure key management* – security of encryption keys is maintained.
- *Upgradeability* – the encryption algorithm may be changed or modified in future versions of the protocol

The security protocol is intended to ensure that data is known only by the source and destination clients. It does not provide non-repudiation; that is, receipt of a message does not prove irrevocably that it came from the apparent sender. The security protocol also does not provide protection against monitoring the volume of data exchanged, and it does not protect against attacks that disrupt the network through the use of spurious control messages.

2.7.1 Encryption algorithm

The baseline encryption algorithm for the PL network is the Data Encryption Standard (DES) operating as a stream cipher in output feedback mode. The keystream is applied only to the payload bits so indicated in section 3.

Each client indicates which encryption algorithms it supports in the beacon payload. A receiving client can indicate which algorithm it desires the source client to use through the PLLC payload, but it must select an algorithm supported by that client.

2.7.2 Procedures for encryption key management

All clients that are members of the same logical network must use the same encryption key. It is intended that key changes should be an infrequent event. The key is most typically provided to the MAC layer through a *MAC-KEY.req* primitive from the client host. In some circumstances, client devices may lack an input/output interface suitable for key entry by the end user. In this case, a means is provided by which the key can be received from another client via the physical medium.

A client device that lacks an end user I/O capability must have a hard programmed key when entering the network. This key typically differs from the key used by other members of the network, and is used only to enable the device to receive the key in use by the other members of the network. The client always retains this key.

The client device needing the network key (the "receiving client") obtains it through the receipt of an encryption key update payload from any other device (the "originating client") in the network. This payload contains the key in use by the network encrypted using the hard programmed key for the receiving client using the initialization vector contained in the key update payload. To minimize the probability of FEC decoder error erroneously causing a key update, a 32 bit CRC is included in the payload. The CRC is formed over the entire payload prior to encryption using the CRC generator described in section 2.3.1. The encryption is applied to the key field and the CRC field.

The receiving client acknowledges the receipt of the key by transmitting a key update acknowledgment payload. To form this payload, the client chooses a new initialization vector, fills the key field with its hard programmed key and computes a new CRC over the entire payload. It encrypts the key field and the CRC using the IV returned in the key update acknowledgment and the key that it received in the key update message. Henceforth all encrypted fields are encrypted using the key received in the key update message.

If the originating client does not receive the key update acknowledgment, it must re-send the key update after each beacon that it receives from the receiving client. If the receiving client has a network key but receives another key update message, it must replace the network key in use with the network key in the key update message and acknowledge the key update message as described above.

The key update payload also contains the Logical Network Identifier in use by the logical network. The receiving client accepts the LNI only if the CRC passes.

2.7.3 Procedures for encryption synchronization

The keystream is initialized through the use of the initialization vector (IV), which is provided in the PLLC payload. While the encryption key is common to all clients in the same network, the IV is unique for each point to point link in the network. Each client must maintain the state of an encryption keystream generator for each other client in the network. When it receives a PLLC

message with a new IV, a client must immediately update its encryption state machine with this new IV.

Updating the initialization vector should be an infrequent event. It must occur on network entry and whenever the receiving client determines that keystream synchronization may be lost. The client may determine that the initialization vector should be updated for other implementation dependent reasons as well.

When a source client receives a new encryption initialization vector from a destination client, it must reset (i.e. set to 0) its sequence numbers for transmission of unicast data frames to that client.

2.8 Procedures for requesting and transmitting test messages

Any client may request any other client in the network to send a test payload. The test payload can be used to evaluate the physical medium for the purposes of selecting payload format parameters, or it can be used to gather performance statistics for network management purposes.

A client requests a test packet by transmitting the test packet request payload. The addressed client responds with a test packet transmitted according to the procedures for contention based access. The data field of the test packet is filled with unencrypted pseudo random data generated using the linear feedback shift register sequence shown in Figure 12. The shift register is seeded with the seed value provided in the received test packet request payload, with the lsb of the seed being loaded into s0 and the msb into s11.

The test packet is encoded and modulated according to the payload format specified in the test packet request payload.

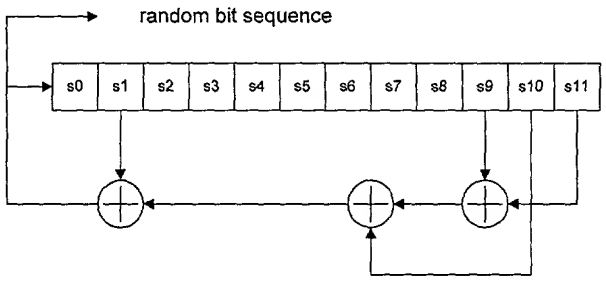

Figure 12 Random sequence generator for test mode

2.9 Procedures for controller based reservation access

To be developed, pending.

3 Packet formats

There are three basic packet types used by the PL network. These are:

- Contention access packets
- Reservation access packets
- Acknowledgment packets

Each of these packet types is described below.

3.1 Contention access packets

3.1.1 Contention access packet format

Contention access packets are used for the transmission of data whose arrival time from the higher layers is unpredictable by the PL network MAC layer. The format of a contention access packet is shown in Figure 13. It begins with a preamble sequence (see phy layer specification). The last symbol of the preamble sequence provides the phase reference for the tones in subsequent OFDM symbols.

After the reference symbol is a segment of header information intended to be received by all clients on the network. This header information is transmitted in ROBO mode, with FEC coding and interleaving that spans only this header portion of the packet, so that other clients can decode the header even if they can not decode the payload. The ROBO mode header field has a fixed length.

Following the ROBO mode header is a payload header and data field. The payload header and data field can carry multiple payloads concatenated one after the other. The payload segment has length and payload format described by the ROBO mode header. The maximum length of this field is 255 OFDM symbols (this is sufficient to transport a maximum length Ethernet frame in most of the coding/modulation possibilities with an allowance for tones not used). Payloads requiring more OFDM symbols than 255 must be segmented and transmitted in multiple packets.

Figure 13 Contention access packet format

3.1.1.1 ROBO mode header field

The ROBO mode header field is shown in Figure 14. There are 20 bits in the ROBO mode header. The meanings of the various fields are as follows:

- *SS* – an index that determines the seed to be used to initialize the scrambler.
- *Payload format index* – the index to the payload format to be used by the recipient of the packet.
- *Payload length* – the number of OFDM symbols that make up the payload portion of the packet. The bits of payload length in octet 9 are the msbs of the payload length. The maximum allowed payload length is 255 OFDM symbols. Packets that can not be fit into this length must be segmented.
- *PV* – an identifier of the version of the protocol used by the transmitting client. For this protocol, PV should be set to zero. If PV is set to a value not understood by the client, the client must ignore the contents of the payload.

ROBO mode header

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload format index						SS	
octet 2	payload length in OFDM symbols							
octet 3	not used						PV	

Figure 14 ROBO mode header field

3.1.1.1.1 ROBO mode header formatting

The ROBO mode header must be encoded and modulated as follows.

The four unused bits of the third octet are filled with zeroes.

The three header octets are encoded with a T=8 Reed Solomon code as described in section 3.5.1. The four zeroes used to pad the third octet are discarded after encoding, yielding a total codeword length of 84 bits. The lsb of the codeword is denoted c[0] and the msb is denoted c[83].

A 3x bit repetition and interleaving scheme is used in forming the ROBO mode header. The interleaving approach divides the set of available (i.e. unmasked) tones into three equal sized sets and transmits one of the three copies of the codeword in each tone set. The second and third copies are circularly shifted in time so that noise events that are impulsive in the time domain will typically only impact one of the three copies of a given bit. The detailed requirements for this interleaving follow.

Let A be the number of unmasked tones, a value less than or equal to 84.

Compute n = 3*int(A/3), where int(x) is the greatest integer less than or equal to x. n/3 is the number of tones available to each of the three repeated versions of the codeword, assuming that an equal number of tones are assigned to each repetition. Thus there are A-n unmasked tones that are not used by the codeword. The bit values assigned to these tones are set to zero.

Compute the number of OFDM symbols, s, required to represent the codeword with a 3x repetition. s = ceiling(3*84/n), where the function ceiling(x) returns the smallest integer greater than or equal to x.

Extend the codeword c[0]-c[83] to c[s*n-1] by setting c[84+e]=0 for $0 \le e < p$. The value of p is s*n – 84.

Compute an offset value k = s*n/3.

Define an interleaver memory as the array m[row][col], where row is the row address which ranges from 0 to s-1, and col is the column address which ranges from 0 to 3*n-1. Fill the interleaver memory with codeword bits as follows:

For $0 \le k < n$ and $0 \le r < s$, m[r][k] = c[k+r*n].

For $n \le k < 2n$ and $0 \le r < s$, m[r][k] = c[mod(k-n+r*n+int(s*n/3),s*n)].

For $2n \leq k < 3n$ and $0 \leq r < s$, $m[r][k] = c[\mathrm{mod}(k-2n+r*n+\mathrm{int}(2*s*n/3), s*n)]$.

The function mod(x,y) denotes x modulo y.

Each OFDM symbol of the ROBO mode header is then formed from the bits in a row of the interleaver memory, assigning each successive bit of the row to the next available tone in the OFDM symbol. The interleaver row length may be one or two bits shorter than the number of tones available in the OFDM symbol, depending on the number of tones that are masked. In this case, the bit values for the remaining tones must be set to zero.

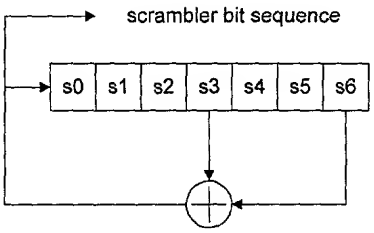

Figure 15 Randomizer block diagram

3.1.1.1.2 Scrambler and scrambler seed

The scrambler used to randomize the data is shown in Figure 15. The contents of the shift register are seeded with the appropriate value from the table of Figure 16 where the msb of the seed is entered in s6 and the lsb in s0. The first bit of the scrambler bit sequence used to cover data is the first bit clocked into s0. The scrambler sequence is applied to all bits in the packet (the ROBO mode header and all payload bits) beginning with the lsb of the payload format index field in the ROBO mode header. The scrambler sequence is not applied to the preamble or the scrambler seed field of the ROBO mode header.

scram seed index	seed value
0	3
1	31
2	55
3	67

Figure 16 Mapping of scrambler seeds to index

3.1.1.2 Payload field

The payload field is encoded and modulated using the parameters specified in the ROBO mode header field.

There are several different types of payloads which are described in greater detail below. A client may transmit more than one payload type in a single packet as long as all the payloads have the same destination. When multiple payloads are transmitted in a single packet, the packet is formed by concatenating the payloads one after the other. (For example, a single packet could carry both an Ethernet data payload and a PLLC payload.) The length specified in the ROBO mode header is the length of the concatenated payload.

The payload can be used to carry MAC layer management payloads and/or unicast data payloads bearing link layer data.

In this section the general format of the payload field is described. Subsequent sections describe the format of specific payload types.

The first octet of a payload is always the payload type. This field is typically followed by a payload header whose format is implicit from the payload type. The length of a given payload is always either implicit from the payload type or explicitly specified in the body of the payload.

The conventions used in this specification are that less significant bits of a field appear at the right of the octets in the payload, and then when a field requires multiple octets, the lsb appears in the first octet and the msb in the last. When a payload is converted to a serial bit stream, lower numbered octets are sent first, and the rightmost bit of an octet is sent first.

Generic payload format

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type							
octet 2-n	data							

Figure 17 Generic payload format

3.1.1.2.1 Payload formatting

The steps involved in formatting the payload portion of the packet prior to modulation are as follows:

- Extending the payload with all ones
- Exclusive-or with the scrambler sequence
- Reed Solomon encoding
- K=7 convolutional coding
- Interleaving

It will often be the case that the payload(s) to be formatted do not fit into an integer number of OFDM symbols. In this case, the payload is extended by appending bits set to 1 until the length is sufficient to fill the last OFDM symbol. If the chosen coding does not allow the padding to exactly fill the last OFDM symbol, then the payload is extended by filling with ones to produce the longest codeword possible that does not extend beyond the last OFDM symbol, and the remaining bit positions in the last OFDM symbol are filled with uncoded bits set to one.

The scrambler process is as described in section 3.1.1.1.2.

The scrambled payload data (which may consist of the concatenation of multiple payloads) is formed into one or more Reed Solomon codewords. The payload data is used to form as many unshortened codewords as possible, starting with the first bit of the payload. The last codeword is shortened if necessary.

The Reed Solomon code words are then converted to a serial bit stream for convolutional encoding.

3.1.2 Payload types

3.1.2.1 Contention access payloads

3.1.2.1.1 Management payloads

Management payloads are used for the control and monitoring of the network. The management payload types are:

- Beacon
- Physical layer link control
- Reservation establishment, acknowledgment and renewal
- Test packet request
- Test
- Medium blanking
- Encryption key update, encryption key update acknowledgment

3.1.2.1.1.1 Beacon payload

Beacon payloads are transmitted in broadcast packets using ROBO mode.

The beacon payload is shown in Figure 18. The fields of this payload are defined as follows:

- *Current system time* – the system time in microseconds at the start of the first OFDM symbol of the preamble of the packet containing the beacon payload as understood by the transmitting client. If system time is not known, the client sets this field to all zero.

- *Source address* – the 48 bit MAC address of the client transmitting this beacon payload

- *Logical network identifier* – a sixteen bit identifier for the logical network of which the client is a member

- *Start of next blanking time* – the start of the next blanking time in microseconds counted from the start of the first OFDM symbol in the preamble of the packet containing the beacon payload.

- *Duration of the blanking time* – the interval of time (in microseconds) in which v1.0 clients are prohibited from transmitting.

- *Duration of the v1.0 time* – the interval of time (in microseconds) after the blanking time in which v1.0 devices may access the physical medium. The sum of this parameter and the duration of the blanking time is the period of the blanking process, and the pattern of blanking time and v1.0 time repeat with this period until the blanking structure is changed.

- *Max_slots* – the max_slots parameter used by the DEFER state for contention resolution after entry from either the SUSPEND IDLE or SUSPEND DEFER states.

- *Logical distance* – a field that indicates the logical distance from the non-v1.0 device that has established the blanking parameters (see section 2.2).

- *e* – set to 1 if an extension field is present, set to 0 otherwise

- *Supported mod types* – indicates which modulation types (as defined in the PLLC payload) are supported by the source client. Setting bit n of this octet to 1 indicates that mod type n is supported.

- *Supported encryption algorithms* – indicates which encryption algorithms (as defined in the PLLC payload) are supported by the source client. Setting bit n of this octet to 1 indicates that encryption algorithm n is supported.

- *Extension field* – additional fields that can be used to represent other upgraded capabilities of this client to be defined in subsequent protocol revisions. The first octet of the extension field is the length of the remainder of the extension field in octets. The contents of the extension field (other than the length) must be ignored by v1.0 clients. The extension field is only present if the e extension bit is set.

If no blanking profile is known, then all bits in octets 8-20 are set to 0.

Beacon packet payload

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type = 0							
octet 2-7	Source Address							
octet 8-11	logical network identifier							
octet 12-15	current system time in usec							
octet 16-18	start of next blanking time in usec relative to current time							
octet 19-21	duration of blanking time in usec							
octet 22-24	duration of v1.0 time in usec							
octet 25	max_slots							
octet 26	spare				e	logical distance		
octet 27	supported mod types							
octet 28	supported encryption algorithms							
octet 29-n	extension field							

Figure 18 Beacon payload fields

(Editorial note: The beacon payload could also specify all the other clients that the transmitting client knows about using a distance scheme.)

3.1.2.1.1.2 Physical layer link control payload

The physical layer link control (PLLC) payload (see Figure 19) is a unicast payload used to specify the payload format that the source client desires the destination client to use when it transmits payloads to the source client.

Although the PLLC payload is a unicast payload, the payload format information contained within it may be used by any client that can successfully receive the packet. Each payload format index references a particular set of enabled tones modulated with the specified modulation and encoded with the specified FEC coding. Payload formats corresponding to payload format index 0 through 7 are predefined and may not be defined in a PLLC payload.

The PLLC payload also can be used to transmit an encryption initialization vector to the destination client. Only the destination client may use this initialization vector.

The fields of this payload are defined below:

- *Source address* – the 48 bit MAC address of the client transmitting this PLLC payload
- *Destination Address* – the 48 bit MAC address of the intended recipient of the PLLC payload.
- *Payload format index* – the index for the payload format that the destination client must use when transmitting payloads to the source
- *iv* – when set to 1, indicates that an encryption initialization vector field will follow
- *pf* – when set to one, the pf bit indicates that this PLLC payload will include one or more payload formats. If zero, the payload contains no payload formats. If the iv bit is set, then the payload format information follows the encryption information. Otherwise it appears beginning with octet 16.
- *Spare* – these bits should be set to 0 by v1.0 devices transmitting the payload and should be ignored by v1.0 devices receiving the payload
- *Encryption algorithm* – specifies the algorithm used for encryption. 0 is no encryption, 1 is 56 bit DES in output feedback mode. All other values are undefined. This field is present only if the iv bit is set.
- *Encryption initialization vector* – 64 bits of encryption initialization vector (IV). If the encryption algorithm requires less than 64 bits, the bits used to form the initialization vector are taken starting with the lsb of the first octet of the IV. This field is present only if the iv bit is set.
- *Payload format index* – the payload format index field in any octet after octet 14 is the index used to reference the payload format specified by the code parameters in the same octet and the modulation type and tone map of the subsequent 11 octets. Payload format index values from 0 to 7 are reserved for predefined payload formats. Payload format index values from 8 to 255 may be defined in the PLLC payload.
- *C* – the convolutional code rate to be used in encoding the payload of unicast packets destined for the sender of the PLLC payload. 0 = rate ½ k=7, 1 = rate ¾ k=7, 2 = no convolutional coding. Others are undefined and reserved for future expansion.
- *T* – the number of symbols corrected by the Reed Solomon code to be used in encoding the payload of unicast packets destined for the sender of the PLLC payload. 0 = 4 errors corrected, 1 = 8 errors corrected. Others are undefined and reserved for future expansion.

- *Mod type* – the modulation type to be used in formatting the payload of unicast packets destined for the sender of the PLLC payload. 0 = ROBO, 1 = DBPSK, 2 = DQPSK, 3 = D-8PSK, 4 = D-16 star QAM, 5-7 reserved. Types 0, 1 and 2 are mandatory. Types 3 and 4 are optional.

- *m* – (more) if set to 1, there is another payload format following this one. Otherwise, set to 0.

- *Tone map data* – the 84 bits of this field (tm0-tm83) specify that tone n is active when tmn=1 and inactive otherwise. For the nth payload format contained in this PLLC (assuming iv=1), tm0 is the lsb of octet 13n+13 and tm83 is the msb of octet 13n+23.

Physical layer link control payload

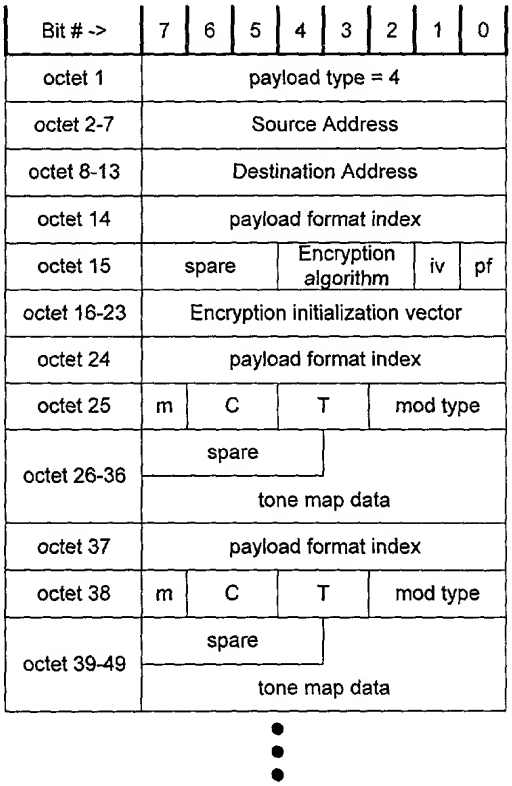

Figure 19 PLLC payload field

3.1.2.1.1.3 Reservation establishment, acknowledgement and renewal payloads

The reservation establishment (RE), reservation acknowledgment (RA) and reservation renewal (RR) payloads (see Figure 20) are broadcast payloads used to establish and maintain a circuit connection between two clients. They must be transmitted in ROBO mode. The procedures for using these payloads are described in section 2.4.

The fields of the RE, RA and RR payloads are as follows:

- *Payload type* – set to 5 for RE, 6 for RA and 7 for RR.

- *Source Address* – the 48 bit MAC address of the client transmitting this payload

- *Destination Address* – the 48 bit MAC address of the client with whom the circuit connection is to be established.

- *Reservation lifetime* – the duration of the reservation in units equal to the reservation period. When the lifetime has elapsed, the reservation must either be renewed or terminated. In an RR payload that terminates a reservation, the lifetime is set to zero and the subsequent fields are not present.

- *Reservation start time* – the time in microseconds from the start of the first symbol of the preamble of the packet bearing the RE payload to the start of the first symbol of the preamble of the packet to be sent in the first reserved time. This field is not present in an RR payload with lifetime set to zero.

- *Reservation period* – the time in microseconds from the start of the first symbol of the preamble of one packet in the reservation to the start of the first symbol of the preamble of the next packet in the reservation. This field is not present in an RR payload with lifetime set to zero.

- *Forward reservation length* – the length of the forward packet to be transmitted each reservation period in OFDM symbols. This field is not present in an RR payload with lifetime set to zero.

- *Reverse reservation length* – the length of the reverse packet to be transmitted each reservation period in OFDM symbols. If the reservation is one way, this field is set to 0. This field is not present in an RR payload with lifetime set to zero.

- *Status* – Status of the reservation. Set to zero to denote normal condition. Other conditions are described in section 2.4.

Reservation establishment (RE)
reservation acknowledgment (RA) and reservation
renewal (RR) payloads

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type = 5 (RE), 6 (RA) or 7 (RR)							
octet 2 - 7	Source Address							
octet 8-13	Destination Address							
octet 14	Reservation lifetime (periods)							
octet 15-16	Reservation start time							
octet 17-19	Reservation period							
octet 20	Forward reservation length (OFDM symbols)							
octet 21	Reverse reservation length (OFDM symbols)							
octet 22	status							

Figure 20 Reservation establishment, acknowledgment and renewal payloads

3.1.2.1.1.4 Status payload

- *Source Address* – the 48 bit MAC address of the client transmitting this payload
- *Destination Address* – the 48 bit MAC address of the client to which this test packet request payload is addressed.
- *Last good sequence number* – the sequence number of the last complete and in-sequence frame received from the destination client.
- *Status code* – a code to indicate the status of the receiver with respect to the destination client. 0 = ready, 1 = out of sequence, 2 = need encryption initialization vector, 3 = need payload format index. Other values are not defined and should be ignored.

Status payload

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type = 10							
octet 2-7	Source Address							
octet 8-13	Destination Address							
octet 14	last good seq. no.				status code			

Figure 21 Status payload

3.1.2.1.1.5 Test packet request payload

A test packet request payload is used to cause the recipient to transmit a test packet, and specifies the length and format of the test payload of that packet. The format of the test packet request payload is shown in Figure 22. The fields are defined as follows:

- *Source Address* – the 48 bit MAC address of the client transmitting this payload
- *Destination Address* – the 48 bit MAC address of the client to which this test packet request payload is addressed.
- *Payload length* – the number of OFDM symbols that make up the payload portion of the test packet to be transmitted by the recipient of the test packet request payload. The bits of payload length in octet 9 are the msbs of the payload length.
- *Payload format index* – the index to the payload format to be used in modulating the test packet to be transmitted by the recipient.
- *Random data generator seed* – the initial seed value for the random number generator that the destination client uses to form the test payload. The lsb is bit 4 of octet 10 and the msb is bit 7 of octet 11.

Test packet request payload

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type = 2							
octet 2-7	Source Address							
octet 8-13	Destination Address							
octet 14	payload length in OFDM symbols							
octet 15	payload format index							
octet 16	random data generator seed							
octet 17								

Figure 22 Test packet request payload

3.1.2.1.1.6 Test payload

The test payload (shown in Figure 23) is transmitted by a client in response to a test packet request payload. It is formatted and modulated according to the payload format specified in the test packet request payload. The data field of the test payload consists of pseudo random data generated using the linear feedback shift register shown in Figure 12. The fields of this payload are:

- *Source Address* – the 48 bit MAC address of the client transmitting this payload
- *Destination Address* – the 48 bit MAC address of the client to which this test packet request payload is addressed.
- *Pseudo random data* – the test data generated by the linear feedback shift register.

Test payload

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type = 3							
octet 2-7	Source Address							
octet 8-13	Destination Address							
octet 14-n	pseudo-random data							

Figure 23 Test packet payload

3.1.2.1.1.7 Medium blanking payload

The medium blanking payload is transmitted in a broadcast packet and allows a non-v1.0 device to confine transmissions of v1.0 devices to a limited time period. The medium blanking payload fields are shown in Figure 24.

- *Source Address* – the 48 bit MAC address of the client transmitting this payload

- *Current system time* – the system time in microseconds at the start of the first OFDM symbol of the preamble of the packet containing the medium blanking payload. The reference for the system time is determined by the source client.

- *Start of next blanking time* – the start of the next blanking time in microseconds counted from the start of the first OFDM symbol in the preamble of the packet containing the medium blanking payload.

- *Duration of the blanking time* – the interval of time (in microseconds) in which v1.0 clients are prohibited from transmitting.

- *Duration of the v1.0 time* – the interval of time (in microseconds) after the blanking time in which v1.0 devices may access the physical medium. The sum of this parameter and the duration of the blanking time is the period of the blanking process, and the pattern of blanking time and v1.0 time repeat with this period until the blanking structure is changed.

- *Max_slots* – the max_slots parameter used by the DEFER state for contention resolution after entry from either the SUSPEND IDLE or SUSPEND DEFER states.

The non-v1.0 client must transmit this payload in ROBO mode with the protocol version of the ROBO header field to indicate 1.0.

Medium blanking payload

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type = 1							
octet 2-7	Source Address							
octet 8-11	current system time in usec							
octet 12-14	start of next blanking time in usec relative to current time							
octet 15-17	duration of blanking time in usec							
octet 18-20	duration of v1.0 time in usec							
octet 21	max_slots							

Figure 24 Medium blanking payload

3.1.2.1.1.8 Encryption key update payload

The encryption key update payload is used to provide the current network encryption key to a client that lacks an end-user input/output capability. There are two payloads used for this purpose, a key update payload and a key update acknowledgment. These are shown in Figure 25. The fields are defined as follows:

- *Payload type* – set to 8 for key update and to 9 for key update acknowledgment.

- *Source Address* – the 48 bit MAC address of the client transmitting this payload.
- *Destination Address* – the 48 bit MAC address of the destination client.
- *Logical Network Identifier* – the 32 bit identifier for the network that the client is to become part of.
- *Encryption initialization vector* – the initialization vector to be used for deciphering this payload.
- *Spare* – these bits should be set to 0 by v1.0 devices transmitting the payload and should be ignored by v1.0 devices receiving the payload
- *Encryption algorithm* – specifies the algorithm used for encryption. 0 is no encryption, 1 is 56 bit DES in output feedback mode. All other values are undefined.
- *Encryption key* – if the payload type is 8, this field contains the 64 bit encryption key used by the network. If the payload type is 9, this field contains the hard programmed encryption key for the device. If the encryption algorithm requires fewer than 64 bits, then bits of the actual key appear first in the field and the positions for unused bits must be filled with bits generated by a random number generator. This field is encrypted.
- *Frame checksum* – a 32 bit CRC computed over the entire payload starting with the payload type. The frame checksum is computed on the unencrypted payload.

The encryption key field and the frame checksum must be encrypted. All other fields must be transmitted in the clear.

Encryption key update and update ack payload

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type = 8 (update) or 9 (update ack)							
octet 2-7	Source Address							
octet 8-13	Destination Address							
octet 14-17	Logical Network Identifier							
octet 18-25	Encryption initialization vector							
octet 26	spare						Encryption algorithm	
octet 27-34	Encryption key							
octet 35-38	Frame checksum							

Figure 25 Encryption key update and update ack payload

3.1.2.1.2 Unicast data payloads

3.1.2.1.2.1 Ethernet data payload

Ethernet data is transmitted using the payload format of Figure 26. The fields of this payload are defined as follows:

- *Payload type* – set to 16 for Ethernet data payloads.
- *Source Address* – the 48 bit MAC address of the client transmitting this payload.
- *Destination Address* – the 48 bit MAC address of the destination client.
- *Length* – The length n in octets of the entire payload
- *Sequence number* – a sequence number that increments module 16 for each new network layer frame transmitted.
- *Segment number* – a number identifying which segment of the frame is transmitted. Procedures for setting the segment number are defined in section 2.5.
- *Data* – the payload data. Includes the ethertype, data and frame checksum fields of the Ethernet frame. The source and destination addresses are not included. The encryption keystream is applied to all bits of this field.
- *Checksum* – a 16 bit CRC computed over octets 2 through n-2 of the payload.

Ethernet data packet payload field

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type = 16							
octet 2-7	Source Address							
octet 8-13	Destination Address							
octet 14-15	Length (octets)							
octet 16	sequence no.				segment no.			
octet 17 to n-2	data							
octet n-1 to n	checksum							

Figure 26 Ethernet data payload

3.1.2.1.2.2 Generic unicast data payload

Other types of network layer data may be transmitted using the generic unicast data payload format of Figure 27. The fields of this payload are defined as follows:

- *Payload type* – set to 17 for generic unicast data payloads.
- *Source Address* – the 48 bit MAC address of the client transmitting this payload.
- *Destination Address* – the 48 bit MAC address of the destination client.
- *Length* – The length n in octets of the entire payload

- *Sequence number* – a sequence number that increments module 16 for each new network layer frame transmitted.
- *Segment number* – a number identifying which segment of the frame is transmitted. Procedures for setting the segment number are defined in section 2.5.
- *Data* – the payload data. The encryption keystream is applied to all bits of this field.
- *Checksum* – a 16 bit CRC computed over octets 2 through n-2 of the payload.

Generic unicast data packet payload field

Bit # ->	7	6	5	4	3	2	1	0
octet 1	payload type = 17							
octet 2-7	Source Address							
octet 8-13	Destination Address							
octet 14-15	Length (octets)							
octet 16	sequence no.				segment no.			
octet 17 to n-2	data							
octet n-1 to n	checksum							

Figure 27 Generic unicast data payload

3.2 Reservation access packets

Reservation access packets are used when the MAC provides a point-to-point periodic reservation-based circuit connection with another client. These packets do not require any header information since the source, destination, length and formatting have already been negotiated in the process of establishing the reservation. The packet consists of the same preamble used for contention mode packets followed immediately by payload link layer data. No MAC or phy layer headers are included in this packet. The entire payload data field is encrypted.

preamble	payload data

Figure 28 Reservation access packet

3.3 Acknowledgment packets

Acknowledgment packets (see Figure 29) are used to acknowledge the successful receipt of a contention access packet. The acknowledgment consists of a preamble followed by an encoded and interleaved source address. If no tones are masked, then the source address field requires 4 OFDM symbols.

preamble	source address field

Figure 29 Acknowledgment packet format

The source address field contains the address of the client transmitting the acknowledgment and is encoded as follows. The 48 bit source address is encoded with a T=4 Reed Solomon code according to the procedures of section 3.5.1. The resulting code word has 112 bits. The lsb of the codeword is denoted c[0] and the msb is denoted c[111].

A 3x bit repetition and interleaving scheme similar to that used for the ROBO mode header is used in forming the acknowledgment source address field. The interleaving approach divides the set of available (i.e. unmasked) tones into three equal sized sets and maps one of the three copies of the codeword in each tone set. The second and third copies are circularly shifted in time so that noise events that are impulsive in the time domain will typically only impact one of the three copies of a given bit. The detailed requirements for this interleaving follow.

Let A be the number of unmasked tones, a value less than or equal to 84.

Compute n = 3*int(A/3), where int(x) is the greatest integer less than or equal to x. n/3 is the number of tones available to each of the three repeated codewords, with an equal number of tones is assigned to each repetition. Thus there are A-n unmasked tones that are not used by the source address field. The bit values assigned to these tones are set to zero.

Compute the number of OFDM symbols, s, required to represent the codeword with a 3x repetition. s = ceiling(3*112/n), where the function ceiling(x) returns the smallest integer greater than or equal to x.

Extend the codeword c[0]-c[111] to c[s*n-1] by setting c[112+e]=0 for $0 \le e < p$. The value of p is s*n – 112.

Compute an offset value k = s*n/3.

Define an interleaver memory as the array m[row][col], where row is the row address which ranges from 0 to s-1, and col is the column address which ranges from 0 to 3*n-1. Fill the interleaver memory with codeword bits as follows:

For $0 \le k < n$ and $0 \le r < s$, m[r][k] = c[k+r*n].

For $n \le k < 2n$ and $0 \le r < s$, m[r][k] = c[mod(k-n+r*n+int(s*n/3),s*n)].

For $2n \le k < 3n$ and $0 \le r < s$, m[r][k] = c[mod(k-2n+r*n+int(2*s*n/3),s*n)].

The function mod(x,y) denotes x modulo y.

Each OFDM symbol of the source address field is then formed from the bits in a row of the interleaver memory, assigning each successive bit of the row to the next available tone in the OFDM symbol. The interleaver row length may be one or two bits shorter than the number of tones available in the OFDM symbol, depending on the number of tones that are masked. In this case, the bit values for the remaining tones must be set to zero.

3.4 Preamble

All packets begin with a preamble used by the receivers for carrier sense. The receiver also uses the preamble to determine the initial packet timing and the phase reference for each of the tones in the packet.

3.5 FEC Coding

3.5.1 Reed Solomon coding

The PL network employs Reed-Solomon coding over GF(256). The field generator polynomial for the code is:

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1$$

and the code generator polynomial is

$$g(x) = (x + \mu^0)(x + \mu^1)(x + \mu^2) \ldots (x + \mu^{2T-1})(x + \mu^{2T})$$

where T is the number of correctable errors in the code and 2T+1 is the number of parity check symbols. T can be either 4 or 8 depending on the type of transmission.

If b_0 to b_{n-1} are the n data bytes to be encoded, then the parity check bytes are the coefficients of the remainder of the polynomial division

$$b(x) / g(x) \text{ where } b(x) = (b_o x^{2T+1} + b_1 x^{2T+2} + b_2 x^{2T+3} \ldots + b_{n-1} x^{2T+n})$$

The transmitted code word is formed from the polynomial b(x) + remainder(b(x)/g(x)) with the coefficients transmitted in the order of the power of x in the term in which they appear; that is, the coefficient of the lowest power of x first and the highest power of x last. This means that the parity check symbols appear first, followed by the information symbols.

The coefficients are converted to a serial bit stream for subsequent processing. Coefficients are serialized so that the lsb of a coefficient appears first.

3.5.2 Convolutional coding

The convolutional code is a k=7 code with either rate 1/2 or 3/4. Rate 3/4 is obtained by puncturing the rate 1/2 code.

The convolutional encoder is initialized with all zeroes at the start of the payload portion of a packet. The serial stream of bits from the Reed Solomon encoder are passed through the convolutional encoder one bit at a time. After the last bit of the last Reed Solomon codeword is passed into the convolutional encoder, 6 more zero valued bits are input to the encoder to "tail off" the code to the all zero state.

Details of the convolutional code are in the phy layer specification.

3.6 Tone masks and tone maps

Some of the 84 tones in an OFDM symbol may be masked to prevent their use by the PL network. This may be necessary due to local requirements on emissions at certain frequencies. When a tone is masked, the transmitter makes the carrier at that frequency have zero amplitude. All clients on a common network must use the same tone mask.

Tones may also be declared invalid in the tone map requested by the destination through its PLLC payload. A tone is usually declared to be invalid because the error performance at its frequency is unacceptable. Invalid tones have the same amplitude at the transmitter as any other unmasked tone, but they do not carry data. (In the case that the modulation includes amplitude information, then each invalid tone has amplitude equal to the largest amplitude generated by the modulation.)

The carriers for invalid tones are always modulated to create no phase change from one symbol to the next.

4 Needed corrections

Need to make reservations last through blanking period.

Need to make it so that if there is a collision and clients cannot receive the packet length, they assume maximum length. The maximum length needs to be specified.